CLAIM REJECTIONS – 35 U.S.C. §112

The Examiner's stated first rejection, for claims 2 and 6, may have arisen from a misreading of particular technical terms in the specification.

The first such is the term 'declarative method'. 'Declarative method' has a particular and limiting meaning when, as in the present examination, it applies to both computer programming and business process implementations. As used in the specification, the phrase 'declarative method' refers to an implementation of what is known as the 'declarative paradigm'. Under this paradigm a pre-existing controlling system (respectively, a logic programming computer environment, or a corporation's existing operational hierarchy and operating procedures) combines the explicitly detailed and designed constraints and the delimited model of the real world and then: (a) first uses the in-built solution-finding mechanisms of the controlling system to induce a solution; (b) secondly uses that solution to run the computer program or to perform the business processes; (c) thirdly measures the results of each particular operation against the explicitly detailed and designed constraints; and (d) inductively uses such feedback to modify the solution, until an unacceptable level of performance is observed. From such context and through its in-built mechanisms a particular, optimized resolution which is itself a process emerges. More particularly, the declarative method is based on stating goals or objectives in a testable fashion (i.e., what is to be achieved) rather than precise instructions (i.e., how to achieve the goal). In this manner, the system (whether a logic programming computer environment or a corporation' existing operational hierarchy and operating procedures, constructs and executes a set of instructions that can be responsive to transient constraints and possibly optimized by some other criteria such as minimizing resource usage.

The declarative paradigm differs from the more traditional procedural paradigm in that it provides a mechanism for producing solutions out of constrained models and declared goals, rather than prescribing a particular solution whose applicability is limited by the designer's foresight, specific instructions, and precision. While the original creator of a particular implementation (such as a manger) probably will not be able to predict the precise solution that is induced on any particular instantiation as a result of the

declarative method, any solution (and all individual solutions) which can be inductively discovered are necessarily contained within the constrained possibilities created by the designer.

The Examiner's objection, that 'A declarative method may be construed as simply an explanation of the method', is not well-founded as it erroneously interpreted the technical term. As used in the claim, the declarative method is not and cannot be read to be an explanation of, or statement, of the method used to create the final result. Instead, the declarative method is a well-defined means (or method, or process) for inductively reasoning to a solution, implementing such a solution, measuring the current effectiveness of that solution against the already-stated effectiveness constraints, and as necessary modifying or altering the inductive reasoning to reach an improved solution until the minimal set of operative constraints are met.

This meaning of the term is made more certain, no less definite, by the claim language "reduction to a form of logic". Inductive reasoning is a logical process which operates according to known, well-defined, principles. For the declarative method to work in any computer-based program, the constraints and model of the problem must be defined in logically correct and complete terms. The declarative method and paradigm are in part feasible precisely because they are and require logic in the acceptable form as found in computer applications; in fact, they sponsored a new approach to programming, and the computer languages for it such as PROLOG (which stands for 'Programming in Logic'; see W. F. Clocksin, C. S. Mellish, Programming in Prolog, 2d Edition, Springer-Verlag, 1984, ISBN 3-540-15011-0. As they state: "Many newcomers to Prolog find that the task of writing a Prolog program is not like specifying an algorithm in the same way as in a conventional [i.e. procedural] programming language....The Prolog approach is rather to describe known facts and relationships about a problem, than to prescribe the sequence of steps taken by a computer to solve the problem. When a computer is programmed in Prolog, the actual way the computer carries out the computation is specified partly by the logical declarative semantics of Prolog, partly by what new facts Prolog can "infer" from the given ones, and only partly by explicit control information supplied by the programmer." To those familiar with the art of computer programming, the phraseology is definitive and constraining rather than indefinite. The entire field of

relational databases represent an area where the declarative method, and its instantiation by reduction to a form of logic has been operative for nearly two decades as a distinct and differentiated approach to the procedural programming method. See C. J. Date, An Introduction to Database Systems, 6th Edition (ISBN 0-201-54329-X); C. C. Fleming and Barbara von Halle, Handbook of Relational Database Design; (ISBN 0-201-11434-8); and for one approach commonly used, F. Pascal, SQL and Relational Basics (ISBN 1-55851-063-X).

Similarly, in business process applications, the phrase 'reduction to a form of logic" is equally definite and certain. MBA students perform a number of exercises (some alone, some in teams, all as part of their formal studies) where the bounds and constraints of the problem facing a business entity are declared and stated. These business cases are divorced from the actualities of the particular instantiation in the real world on which they were based; as the names of the companies and individuals involved are often replaced, and the totality of original detail reduced to a symbolic abstraction. Those examining the case need not know all the irrelevant details, and the source problems are stated in terms of a business logic which is both feasible and comprehensible, yet also constrained, to the instantiated 'managers'. Any given implementation or instantiation may be less-than-mathematically pure in its logical form. Most, in fact, are certain to be so (for only computers need specific definitions of the distinctions which may be drawn between 'dollars', 'sense' and 'cents', between 'a yen', and "¥ = Yen"). The reduction produces a level of abstraction which can be resolved by a mechanical operation of the processing entity in the environment, context, and simulation; it does not and cannot include any element of human judgment as to the interpretation of the definitive symbology.

The second such misunderstood term is 'dynamic process'. In the prior art, the term 'dynamic process' refers to a process in which an instantiation thereof is an active transformation of the subject matter of that process; for example, a process for creating boards from trees would involve aligning the trees and three dimensions of cutting blades to produce milled lumber of the proper length, width, and thickness. It could specify which dimension was cut first, whether the trees were examined for incorporated hazards, or even what to do with the build-up of sawdust that might create jams or other hazard.

But that process would not change; only the inputs (trees) would. Each time a tree was run through the mill an instance of the process would be active and a dynamic change would occur to the resource, but the process itself would be static. The example process would not serve well for producing circular tabletops from massive redwood burls.

However, unlike the prior art, the term "dynamic process" as used in the application refers to a process which is itself emerging, self-evaluating, or evolving, rather than one in which an instance of the process is simply "active." Though not used in the claims, the difference has been brought in the specification by the inclusion of the term 'transformative process' for what the Examiner seems to have read from the prior art as a dynamic process, i.e. a process which dynamically transforms inputs. Under the definition of a dynamic process as one in which the process itself is evolving, a dynamic process for creating boards from trees might induce, from the greatly-different dimensional requirements and anticipated inputs, completely new settings for the saws which would adapt to the new inputs and goals, when the sawmill moved from lumber to circular-burl tabletops. The dynamism is focused on the process and its interaction with its internalized model of the world, rather than just on the transformation of the inputs in the real world.

The third such misunderstood term is 'business process'. A business process as it is traditionally described is essentially a transformative process that takes a set of inputs and invokes activities and generates outputs (which may and should include records of the activity and results). Information may be all that is changed (financial accounts, credit terms, even contractual arrangements) but so, too can physical inputs (e.g. the catalytic chemical refinement of crude oil, growth-in-media of a vaccine or other biomedical material production, or the creation of an automobile from plastic, carbon, glass, paint, iron and more). To the extent that the same process changed materials and was used by a business, a transformative process was equivalent to a business process.

However, the management of a transformative process is entirely, and separately, a business process, particular and unique to each entity which engages in such transformations, yet generalizable to a set of instantiations of such transformative processes. How 3M handles each research effort, patent application, and licensing and/or manufacturing-distribution-and-sales effort is a 'business process' which many other

companies have sought to emulate. Alternatively, how PayPal managed auction-payment cooperative funds transfers between random and self-selecting pairs of individuals was a business process that Ebay both considered worth hundreds of millions (as Ebay bought PayPal outright) and had not successfully managed to duplicate. It is the management of transformative processes (which would include employee hiring, retention, measurement, and firing), not just the transformative processes themselves, which are included in the term "business process".

The specification has been amended to incorporate references, examples and materials relevant to and explanatory of the technical terms discussed above which are used in the claims.

Using these definitions, the claims thus explain not how any particular business process operates on its inputs within a model which includes defined outputs, but how to design and implement a truly dynamic process to enable an optimized business process to be inductively created by the underlying constraints, model and goals, without the necessity for human analysis, discovery, and validation for that particular emergent transformative process.

The language of the claims is believed, therefore, to meet the requirements of 35 USC §112, 'pointing out and distinctly claiming the subject matter which the applicant regards as his invention'. However, the applicant has revised his claims to better meet this (and other) of Examiner's objections as explained below.

The Examiner's objection to Claim 5 and 9-11 as 'incorporating a means into the method' has been acknowledged and the typographical replacement of 'means' for 'steps' has been corrected and the claims restated. The same corrective process has re-stated Claim 1, which Examiner did not so criticize, for clarity.

CLAIM REJECTIONS – 35 U.S.C. §102

The Examiner's rejection of Claims 1-12 as being anticipated by Davis et. al., The Information System Consultant's Handbook ("Davis") is respectfully erroneous in its reading of that reference. The crucial, albeit somewhat subtle difference between the present application and the prior art can be stated thusly: Davis discloses a method for managing a dynamic process; while the present application discloses a method for dynamically managing a process, which process itself may manage a subordinate dynamic (i.e. transformational) process. Davis, et. al. talk about building a process (chiefly, an expert system or other computer implementation) that will replicate the procedures encoded into it by the builders; the present invention talks about building a tool that will create or discover new and better expert systems or computer implementations automatically. This invention is, in effect, a way to replace the very Information System Consultants to whom Davis' work was directed. Davis' work can replace a line worker with a 'black box', but this invention can replace *Davis* with a black box.

Each of the steps detailed in the second paragraph on page 6 of the office action ('declaring an objective'; 'stating..a set of rules', 'testing each rule', 'actuating a rule when its condition is met', and 'delegating the objective', is according to Davis performed during the 'Identification Phase' as part of creating an expert system (i.e. once). These steps are performed by the human expert system analysts and/or designers (see p. 51, 2d paragraph).Unlike in the present invention, they are not self-referential; nor are they dynamically responsive; nor do they enable inductive emergence of each of the subordinate steps. Throughout Davis there is an underlying presumption of a one-time development of a thereafter static transformational process; this can be seen in the first sentence in Chapter One of Davis: "The purpose of a methodology is to specify a set of well-defined steps or phases, coupled with a set of clear, measurable exit criteria, for solving a complex problem (such as developing an information system)." Davis teaches a one-off approach which must be implemented by individuals (and those with considerable domain expertise) to obtain a top-down design for a solution, that approach

having to be repeated whenever the resulting design proves inadequate or whenever the world differs sufficiently from the model thereof incorporated into the designed solution.

Moreover, the second paragraph makes it even clearer that Davis' methodology is nothing more than "a memory aid". Davis' methodology cannot be duplicated, though it may be re-enacted from the same context; however, given the variable performance of human beings, is unlikely to produce a like, let alone an identical, result. But the present invention produces a solution set which can be duplicated without having to be recreated; a solution set which can be physically duplicated without requiring comprehension of the dynamics of its creation or operation, merely by reproducing the physical instantiation thereof, just as a book can be duplicated by photoreproduction or a re-run through a printing press far more faithfully and accurately than it could by having the author re-write the book. This alone makes the present invention far more valuable to a business seeking to spread and sustain the knowledge it has learned globally and over time independent of the human workers who interact with the rest of the world.

A principle problem with Davis' prior art is identified in the second full paragraph on page 4: "There is always a concern that the system developed may not accurately reflect the current business environment. The elapsed time between the initial proposal and system completion can be quite lengthy (often one or more years." Because the present invention is operative concurrently with the changes in the business environment and enables a responsive interaction that changes the business process because of both observed and anticipated changes in the real world, it overcomes this major disadvantage of the prior art.

In Davis the closest thing to a process, as it is defined in the present application, is his "methodology" which is well-defined from the beginning "a set of well-defined steps or phases" (p. 3); it is specifically distinguished from a 'process' which is defined in Davis as "an activity that changes the system in some way" (p. 5). This definitional clarity is then muddied by the further statement that "A key purpose of a methodology is ensuring that nothing is overlooked in the process of solving a complex problem" (p. 6). The need for human intervention is made explicit as well with the statement that a methodology is "a body of practices, procedures, and rules used by those who work in a discipline or engage in an inquiry" (p. 6).

One of the major differences between Davis and the present invention, is that Davis lacks, and the present invention provides, an implementation which is an emergent rather than a dictated process. Under Davis the system designers focus on what their tool must do, and to do that, the problems must first be explicitly recognized and stated; see p.6, Section 1.4.2: "An information system is "born" when a problem is recognized....Eventually, a change in the nature of the problem...degrade[s] the value of the system, so it "dies"..." Under the present invention, rather than on focusing on what the overall process "must do" , the system designers focus on what the process "can do", by stating the detailed constraints and operative rules (condition/action combinations), which has far less risk of error in the eventual results, since the process is never going to be designed to exceed what the constraints delimit as possible.

Furthermore, Davis does not equate "methodology" and "process," nor does either the process or the methodology of developing the system or programs which he describes, arise from rules which are available to and modifiable by the users thereof; whatever is created using Davis' approach, remains an inaccessible and unmodifiable "black box" of dictates and operations which are liable to grow increasingly disparate from current business conditions and needs..

An analogy may be of assistance. Davis, et. al., hand the user a single fish. The current invention uses the currently most effective method of fishing (be that by line, net, or even shopping at the local market). The first is useful once, but the second is continuously useful and responsive to changing conditions.

Claims 2 and 3 are, as restated, not inherent in Davis. et.al. for the same reasons as stated above.

Claim 4 is differentiated from Claim 3 and thus the objection is inapplicable.

The Examiner's objections to Claims 5-8 are not well-taken, for in the current invention, the internalization of feedback is precisely that (the feedback is part of the process which is itself the invention and thus is self-referential). Consequently, for the present invention, the feedback is and becomes part of the process itself. This is a clear and distinct difference from Davis, where the feedback (namely, the "prototype

revision") he describes is external to the result. The language which is used on page 52 (the citation given by the Examiner) makes this clear: the concepts are 'refined', the solution space 'reformalized', and the system 'redesigned'; there is no reflexivity, which is a key element of the current invention.

This lack of self-reference can be comprehended by a simple analysis of the phrase used by Davis: "prototype revision". The fact that the revisions occur at the prototype stage, suggests that what differentiates the final result is the lack of any such revision; the 'finished product' is viewed as being complete and unchanging, in Davis' approach. But in the current invention, the implementation is continously and adaptively changing in response to a constantly changing world. There is never a 'final' model or process, there is only 'the most current' version.

The Examiner's objection to Claim 9 (and 12) mistakes a single cycle for the process of continually performing the process which instantiated the single cycle, and mistakes the reaching of a solution for a single problem for the process whereby any number of problems are solved. Davis, at page 50, teaches one to "define the problems the system will be expected to solve". The present invention instead teaches one to define the processes by which all problems must be solved; to find, for any particular problem encountered, the process which is most likely to produce a solution; to apply that process to that problem; and then to measure the result against the desired goals for that solution, in order to modify the problem-solving processes appropriately. Turning Davis' approach into a generalized solution is neither taught by Davis nor an obvious extension thereof; were it otherwise, Davis would not have been selling his book to a class of individuals seeking to implement his approach.

The Examiner's citation, for Claim 10, of Davis, section 7.4.2.6, as proof that the induction of a new method to avoid logical contradiction experienced in the process, is unpersuasive. That section only says that testing and evaluation are used to "verify the reasoning and/or inference process". Verification only identifies whether a logical contradiction is experienced. It is arguable that Davis can be extended to teaching the particular of locating the details of a contradiction. However, no method is stated in the reference for resolving the contradiction. Any production of a new method lacking any logical contradiction in Davis is not 'inherently completed', but instead is explicitely

externalized, during the testing and evaluation phase. Under Davis, the system designers must solve all logical contradictions. The method that they use (elimination of one rule at random, redefinition of a term to prevent the occurrence, instantiation of a special, ad-hoc and non-logical correction are all possible alternatives, none of which are taught in the reference but all of which are equally plausible) is not itself incorporated into what the system designers are building. That is a major difference between Davis and the present invention and one which the reference is silent on.

Examiner's citation, for Claim 11, that "Davis et. al. disclose avoiding altering the delegation above the level in which the logical contradiction occurred" is not supported by the text cited. Section 72.4.2.3 is silent as to delegation, level, or logical contradiction. Examiner's reading into the text that 'a centralized management philosophy inherently avoids altering the delegation above the level the contradiction occurred' is but one interpretation. Equally valid, and indistinguishable from the text cited, is an approach whereby all logical contradictions are handled at the highest level, that of the most centralized part of the system, or an approach whereby all logical contradictions are handled by a separable contradiction-handling sub-portion. Applicant respectfully requests further explanation of how Davis, as cited, teaches the adaptive and locallized approach used in the present invention.

Davis, and the other citation offered by the Examiner, do not teach any of the following, which are all contained in the present invention:

(a) the express linkage of resources (defined as constraints) and rule sets;

(b) the use of sets of condition rules, including potentially contradictory sub-sets, with methods for avoiding such contradictions;

(c) the use of delegation to appropriate and immediately applicable solution spaces part of the overall process with inheritance of conditions and upward passage of consequences, coupled to the relevant set (or sub-set) of rules; or, most importantly,

the fundamental approach of inductive or emergent creation of a solution as opposed to an externally-dictated procedural implementation thereof. It is precisely this last point which enables the self-modifying feedback-driven adaptivity which is the most important and differentiated aspect of the present invention from any of the prior art.